

Movie Recommendation based on Users' Tweets

G. Hemantha Kumar
Dept. of Computer Science,
University of Mysore, Mysore, India

Syedmahmoud Talebi
Dept. Of Computer ,
University of Mysore, Mysore, India

ABSTRACT

In this paper new idea for recommending movies has been designed. This system is based on machine learning algorithm which calculate similarity between user's tweets and scenario of the movies. This method could recommend movies which are more similar to users' interest.

General Terms

This work is classify in Pattern Recognition, Recommender System, Artificial intelligence,

Keywords

Movie Recommendation, recommender System, Text Similarity

1. INTRODUCTION

Recommender Systems(RS) playing an important role in Artificial Intelligence. In more specific they are part of the concept of information filtering system which is applicable for prediction in E-commerce and etc[6]. RS can help user to find their relevant products (texts, friend, items) in the huge information space[6].

RS can generate list of suggestion through collaborative and content based algorithm[2]. In collaborative method suggestion happen either based on user's past behavior or other user activities [4] while in content-based approach suggestion is based on similarity between user profile and item description[11].

One of the great application of RS is recommending movies to users based on their preferences. Nowadays too many movies are available in internet which makes it difficult for user to chose among them therefore recommender systems could help to facilitate these needs.

Most movie recommender systems using collaborative algorithm for recommending movies to user. As a matter of fact in collaborative filtering it recommends based on user's friends in social media or based on the ranking which user gave to some movies. In this paper our main concern is recommendation should happen based on user lifestyle and manner rather than friends preferences.

For this purpose we need history or some data of user. To overcome this cold start, system will connect to user's tweets and retrieve posts. After collecting user post we will extract keywords and remove those unwanted words. Then after doing some analysis system compare these extracted keywords with movies' scenario to calculate which movies are similar to users' posts. Therefore among those list algorithm can select top N first of the list as a recommendation to the user.

In this work a personal research scholar database for movies and Python API for collecting users' post in twitter have been used. Now there is database of users' posts and scenario of each user is available. Using methods to compare similarity between users' post and scenario is the concern of this work.

2. RELATED WORKS

recommender systems are applicable for recommending new items based on user's preferences[3]. RS are specially good in e-commerce because they could recommend new items to user which are most likely to buy based on their personal information[5]. Lawrence et al. 2001[8] introduced a recommender system based on user previous purchase. Most of RS using ratings which collected from previous user to recommend items to the new user. One such method which is very useful nowadays has been designed by Jung et al[9].

Movielens is one of the movie recommender systems. After user sign in for the first time, system ask user to rate some movies, therefor based on this ratings system recommending new movies to user[7].

MovieGen is another work has been done in this area which is using machine learning and cluster analysis. This system takes in users' information and recommend movies based on Support Vector Machine(SVM) models. First system cluster the movies and then raise some questions to understand user preferences and based on that list the recommendation [7].

MoRe (Movie Recommender) is the system has been designed for movie recommendation. It use users' rating which gives for some movies and based on that recommend movies[1].

3. DATA

3.1 user's posts

first of all we need to get user's profile details or user's posts. Therefore with the help of Tweepy - python API for connecting twitter- we have collected 30 users' tweets with all the details. This API has a perfect documentation for developers. Here is the most useful function for our work:

```
***** Tweepy API*****  
import tweepy  
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)  
auth.set_access_token(access_token, access_token_secret)  
api = tweepy.API(auth)  
public_tweets = api.home_timeline()  
for tweet in public_tweets:  
    print tweet.text  
*****End API*****
```

3.2 Movies' Scenario

A collection of 42,306 movie plot summaries and metadata at both the movie level (including box office revenues, genre and date of release) and character level (including gender and estimated age).

4. ALGORITHM

In this section we will devise algorithm which could recommend movies based on input data. In the first step let's consider a very simple case of the system and after that we will improve for better performance.

Here let's consider a very basic view of the algorithm. Let's consider a situation which there is movie database and users' posts and system will calculate similarity between twitter posts and movies' scenario. To calculate similarity between texts there are variety of methods which cosine-similarity(Berry & Kogan, 2010)could be one of them.

Algorithm (I):movierecommendation

Input:

```

user_post//Combining all users' tweets in
//one string variable
Movie_DB//Two dimensional array
// holding movie name
// and it's scenario
    
```

Output:

```
list_of_recommendation
```

Begin:

```

keywords=AlchemyApi.extractkeywords(user_post)
for i=1 to all_no_movies
    arr[i]= cosine_similarity(keywords,Movie_DB[i]
    ['scenario'])
list_of_recommendation=top_10_list(arr)
End
    
```

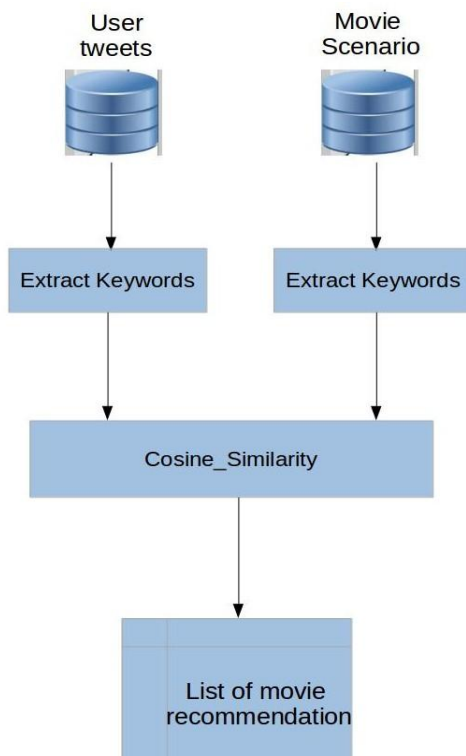


Fig 1: Flowchart of the system

To improve Algorithm(I), it is better to cluster scenarios which save a lot of time. Instead of comparing user_post with all movies it is better to compare the center of cluster. Therefore movies in the cluster could be output of our recommendation.

5. EXPERIMENTAL ANALYSIS

here we will try to discuss practically how algorithm will work. Let's consider a user registered the system. In the first step system try to connect twitter of the user to collect tweets. After removing unwanted words and extracting keywords, system is ready to compare keywords with movies' scenario to calculate similarities.

Here if we think more practically processing above scenario for each user will consume a lot of time. For this purpose it is better to cluster user as well as movies based on their scenario and for each user system will check user belongs to which cluster and suggestion process will much more faster. In any algorithm design time complexity is the main concern therefore here also time complexity has not forgotten. Lets divide the calculation in two parts.

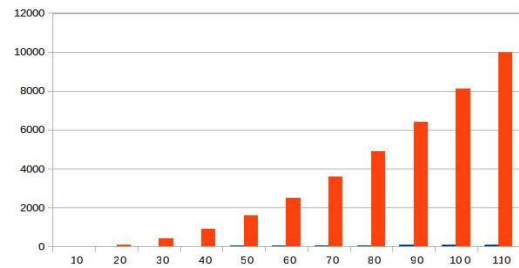


Fig 2: Time complexity for N movies

The first one is clustering movies which has $O(N*N)$ time complexity and for keyword extraction $LOG(N)$. N is the number of movies. Therefore the more movies the more time needed to cluster.

6. CONCLUSION

As it mentioned above most of the movies recommender are based on simple type of recommendation and not trying to get more of the user preferences therefore a method which focus more on users' life style was needed. Utilizing social media for this work made a better result.

7. REFERENCES

- [1] Lekakos G, Caravelas P. A hybrid approach for movie recommendation. Multimedia tools and applications. 2008;36(1-2):55-70.
- [2] Jafarkarimi H, Sim ATH, Saadatdoost R. A naive recommendation model for large databases. International Journal of Information and Education Technology. 2012;2(3):216.
- [3] Lee D, Lee S-K, Lee S. Considering temporal context in music recommendation based on collaborative filtering. In: Proceedings of Korea computer congress. 2009.
- [4] Melville P, Sindhvani V. Encyclopedia of machine learning. Springer-Verlag, chapter Recommender systems; 2010.
- [5] Lee H. Enhancement of Collaborative Filtering in Electronic Commerce Recommender System. Kangwon University Graduation School. Doctoral thesis; 2009.
- [6] Ricci F, Rokach L, Shapira B. Introduction to recommender systems handbook. Springer; 2011.
- [7] Eyjolfsdottir EA, Tilak G, Li N. Moviegen: A movie recommendation system. UC Santa Barbara: Technical Report. 2010;

- [8] Lawrence RD, Almasi GS, Kotlyar V, Viveros M, Duri SS. Personalization of supermarket product recommendations. Springer; 2001.
- [9] Jung S, Harris K, Webster J, Herlocker JL. SERF: integrating human recommendations with search. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management. ACM; 2004. p. 571–580.
- [10] Berry MW, Kogan J. Text Mining. Applications and Theory West Sussex, PO19 8SQ, UK: John Wiley & Sons. 2010;
- [11] Brusilovski P, Kobsa A, Nejdl W. The adaptive web: methods and strategies of web personalization. Vol. 4321. Springer Science & Business Media; 2007.